

Droidride : 使用Android的时候做好安全措施

SYSCAN 2012

Wayne Yan



- 幸好SYSCAN不是在十一举行的

为什么叫droidride?

- Roxette: "Come on, join the **droidride** everybody,
- get your tickets here, step right this way"



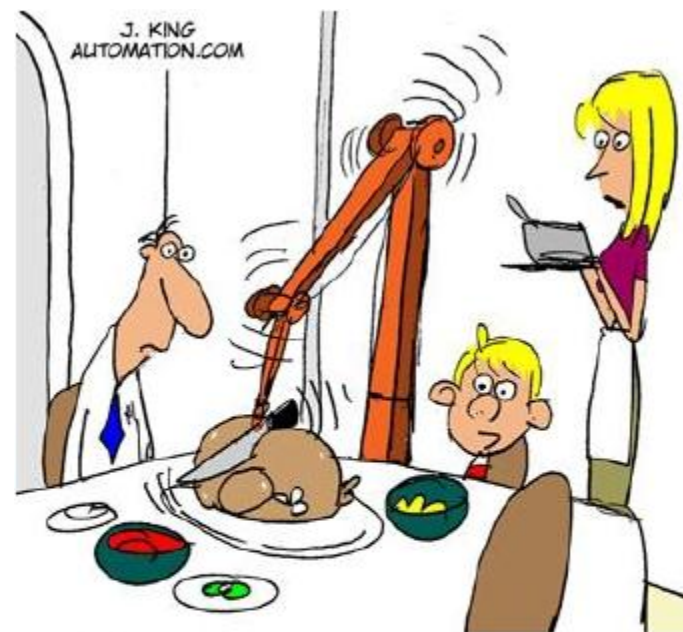
关于我

- 一个罗克塞特乐队粉丝
- 懒人: 恶意代码程序检测粉丝

- 如果有办法自动完成工作,
我从来不会自己动手。
- 乐于寻找有趣的安全程序

一般

Android上的查杀病毒引擎



"The control system is based on my CARVE algorithm...Cut And Render, Verily Eat."

基本问题?

- 2007年的时候，如果病毒特征文件小于100MB
云反病毒程序是否会存在
 - 当然不会
- 如果你不需要单机版的反病毒方案
那么你是否会用基于云的方案
 - 也许是特征、MD5、URL、沙盒等
 - 轻量级的桌面和云结合方案
- 更智能和轻量级的引擎
能大批量有效的检测恶意代码
尤其是对Android平台
是



声明

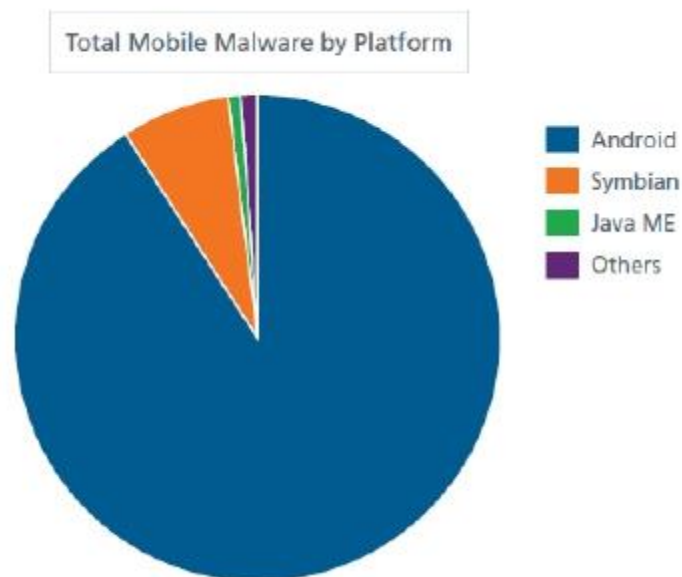
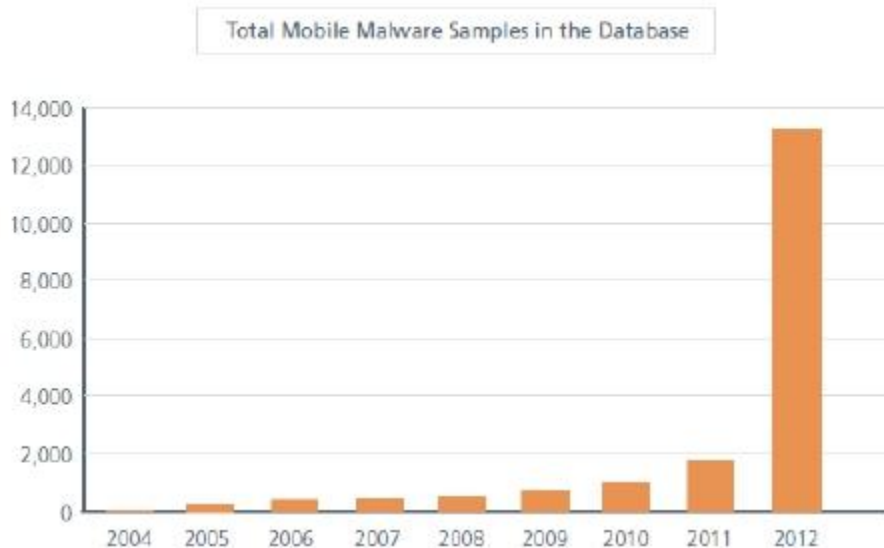
- 个人研究工作
- 不代表我的单位
- 不销售

大纲

- 当前Android遭遇的威胁
- 恶意代码vs. 防御，基于文件，基于流
- Droidride引擎
- 示例
- Q&A

Android威胁现状

- McAfee's 第二季度报告: Android 系统是恶意攻击者首选
- 目前有多少android恶意代码?)
 - 猜测: 120000 – 150000?
 - 白名单: 800000 (一些中国app没有包含在内)?



电脑恶意代码

10 years ago- where?

Number of new samples added to AV-Test.org's malware database (virus collection)

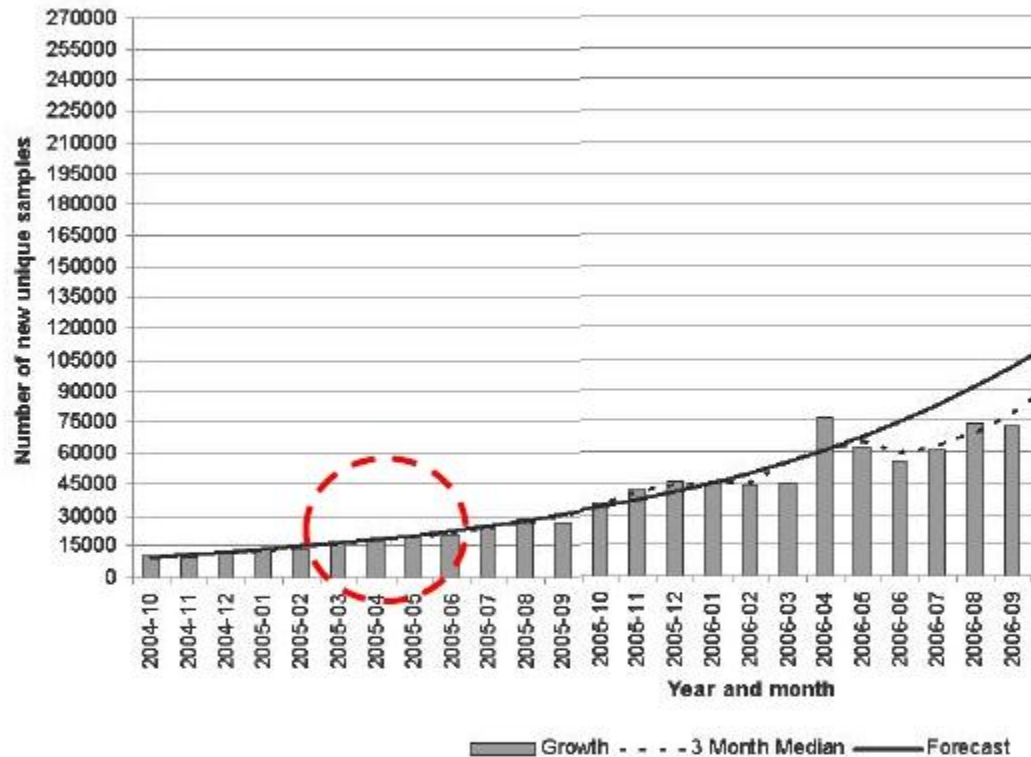
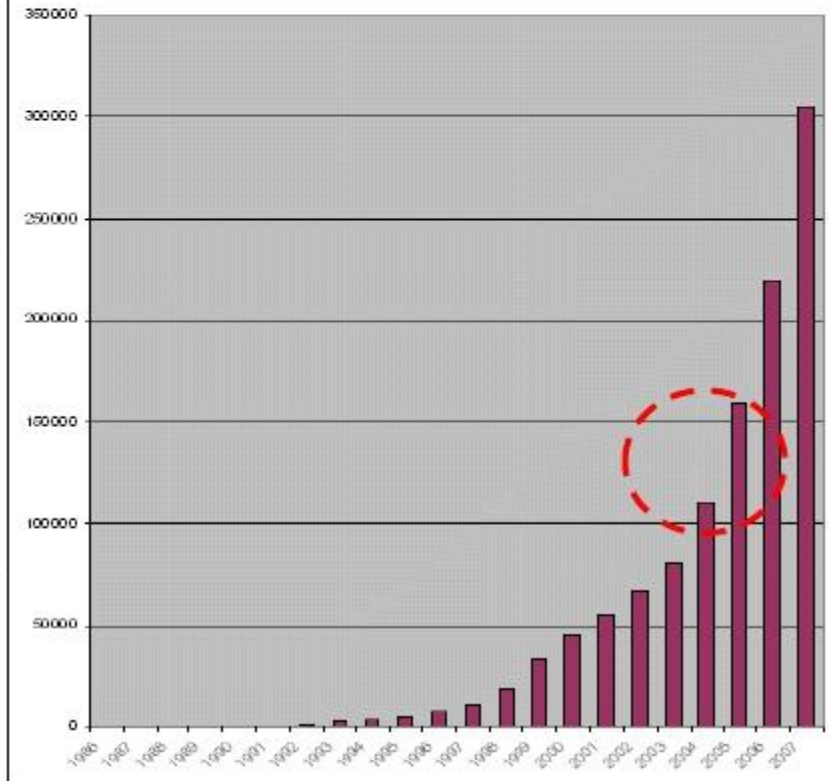


Figure 2: AV-Test's malware collection growth.

Number of viruses 1986-2007



2007年后，电脑反病毒情况

- 反病毒厂商无法检测所有新病毒和新变种
- 检测率从95+%降低到79%
- 用户开始抱怨CPU占用高，导致电脑速度慢
- 每年花费一笔钱在反病毒软件商，但是检测不出来一个恶意软件
(免费拦截所有软件是另一件事了☺)
- 前段产品和后端处理衔接不好
- 新创公司只为利润

电脑反病毒 vs. 手机反病毒

- 你忙着克隆电脑反病毒技术到手机上面么？
- 不幸的是，这样行不通。
 - 2.12年，有41个android反病毒软件，AV-test使用了618个恶意软件，发现其中34个软件检测率低于65%。

From jouret-mobile.be

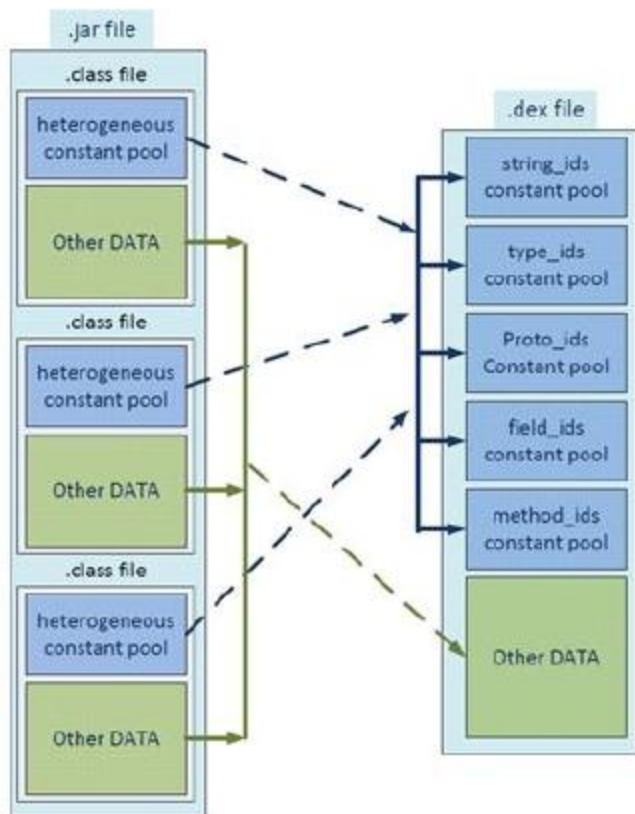
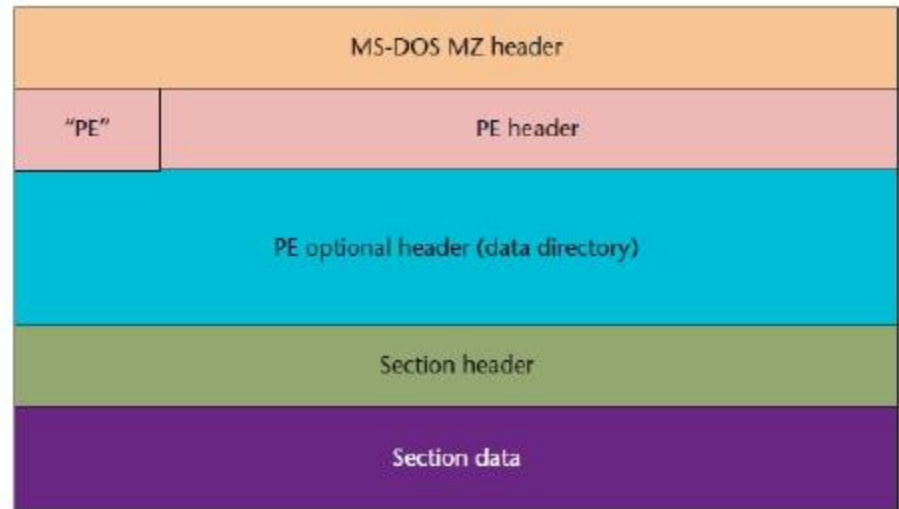


手机上的情况更糟糕

- 电量限制
- CPU限制，不适合采用基于行为的仿真
- 网络带宽
- 简单的混淆就可以避过已有产品的查杀
- 无线网络和有线网络的差别

PE vs. DEX 101

- Windows PE 格式
 - ❑ Data
 - ❑ Resources
 - ❑ Strings
 - ❑ API functions



- Android dex
 - ❑ Strings
 - ❑ Codes
 - ❑ Rigid formats
 - ❑ Work with other files, such as resources and permission file

From <http://imsciences.edu.pk>

■ 手机rootkit (虽然不是大问题)

■ PE 打包vs. 嵌入dex

■ 越来越多的Android打包工具

```
<!DOCTYPE html>
<html>
<head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# githubog: http://ogp.me/ns/fb/githubog#">
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>APKfuscator/resources/apkencrypt.dex at master · strassers/APKfuscator · GitHub</title>
<link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="GitHub" />
<link rel="fluid-icon" href="https://github.com/fluidicon.png" title="GitHub" />
<link rel="apple-touch-icon-precomposed" sizes="57x57" href="apple-touch-icon-114.png" />
<link rel="apple-touch-icon-precomposed" sizes="114x114" href="apple-touch-icon-114.png" />
<link rel="apple-touch-icon-precomposed" sizes="72x72" href="apple-touch-icon-144.png" />
<link rel="apple-touch-icon-precomposed" sizes="144x144" href="apple-touch-icon-144.png" />
```

```
<link rel="icon" type="image/x-icon" href="/favicon.png" />
```

```
<meta content="authenticity_token" name="csrf-param" />
```

```
<meta content="y33b5HsEibgd07Nep7WgZz6/hTb7ooq1GRfn5mzvUI=" name="csrf-token" />
```

```
<link href="https://a248.e.akamai.net/assets.github.com/assets/github-324e543186da6868ee3c68ba953e44a
```

```
<link href="https://a248.e.akamai.net/assets.github.com/assets/github-2a833ca188e6e8e3c5b7de09688a7605
```

Name	Value	Start	Size	Co
int odexpad	0	(local)		
int odex	0	(local)		
char tmp[3]		(local)		
char tmp[0]	32''	(local)		
char tmp[1]	32''	(local)		
char tmp[2]	10	(local)		
struct header_item dex_header		0h	8h	Fg:
struct dex_main magic	<1	0h	8h	Fo:

#	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	64	65	78	0A	30	33	35	00	4C	70	4F	08	F8	8A	70	E3	Dex.D15.jp0.08p8
0010h:	38	2A	18	09	A8	09	91	ED	D1	18	65	4E	3D	40	39	94	0*...w...d...e...t...8*
0020h:	FC	8D	01	00	70	00	00	00	78	56	34	12	00	00	00	00	0...p...x...V...e...
0030h:	00	00	00	00	2C	8D	01	00	72	04	00	00	70	00	00	00	...e...x...p...
0040h:	1F	01	00	00	38	12	00	00	0F	01	00	00	84	16	00	00	...B...e...t...e...t...
0050h:	82	01	00	00	68	23	00	00	99	03	00	00	78	27	00	00	...b#...e...x.../...
0060h:	67	00	00	00	40	4C	00	00	DC	34	01	00	2D	59	00	00	g...8...D4...Y...
0070h:	84	49	01	00	A2	34	01	00	88	68	01	00	5E	5E	01	00	4...o4...x...^...
0080h:	5F	41	01	00	DF	3A	01	00	E5	3A	01	00	07	3A	01	00	_...0...t...&...t...t...
0090h:	39	57	01	00	31	5F	01	00	79	6A	01	00	38	63	01	00	9..._...l...y...j...o...o...
00A0h:	C9	31	01	00	00	4C	01	00	A8	6A	01	00	61	65	01	00	E1...D...j...e...e...
00B0h:	2D	46	01	00	BF	62	01	00	21	63	01	00	50	65	01	00	-F...g...b...c...e...Pe...
00C0h:	38	58	01	00	00	54	01	00	2E	66	01	00	CF	49	01	00	8Z...T...e...f...I...I...
00D0h:	72	37	01	00	17	33	01	00	4F	33	01	00	08	39	01	00	e7..._...o...o...o...o...
00E0h:	C2	39	01	00	67	40	01	00	74	39	01	00	50	27	01	00	A9...q8...t...9...P.../...
00F0h:	96	54	01	00	C5	59	01	00	77	5A	01	00	7C	5A	01	00	-T...A...Y...w...Z...l...Z...
0100h:	AC	62	01	00	DF	68	01	00	CB	59	01	00	5B	66	01	00	-b...0...h...E...Y...[...[...
0110h:	B5	59	01	00	B8	59	01	00	E3	6F	01	00	06	47	01	00	u...y...>...A...o...o...G...
0120h:	D3	35	01	00	E0	31	01	00	3C	32	01	00	F3	3C	01	00	0S...0...1...<...2...6...c...
0130h:	82	28	01	00	6C	6A	01	00	C9	40	01	00	8F	40	01	00	...l...j...E...E...e...E...
0140h:	B8	35	01	00	E3	47	01	00	DF	47	01	00	F9	62	01	00	E5...0...G...>G...0...b...
0150h:	B8	65	01	00	D4	62	01	00	B7	63	01	00	F2	62	01	00	c...e...0...b...b...0...b...
0160h:	83	65	01	00	58	46	01	00	B7	70	01	00	A3	70	01	00	f...e...[...F...p...p...p...p...
0170h:	8A	70	01	00	80	70	01	00	CF	70	01	00	79	70	01	00	8p...0...p...l...p...y...p...
0180h:	09	71	01	00	F5	70	01	00	10	4E	01	00	DF	4E	01	00	...q...0...p...>...H...>A...N...
0190h:	03	47	01	00	0A	49	01	00	9C	6D	01	00	80	6E	01	00	...o...o...I...c...om...e...t...n...
01A0h:	37	72	01	00	5E	46	01	00	C2	67	01	00	C1	53	01	00	7c...^...F...>A...0...A...S...

Name	Value
int odexpad	0
int odex	0
char tmp[3]	dex
struct header_item dex_header	
struct string_id_list dex_string_ids	1130 strings
struct type_id_list dex_type_ids	287 types
struct proto_id_list dex_proto_ids	271 prototypes
struct field_id_list dex_field_ids	366 fields
struct method_id_list dex_method_ids	921 methods

■ DEX 反编译失败

■ 恶意部分可能会藏在其他文件中，而不是dex文件中

Android 打包工具APKfuscator (Strazzere)

- 0-255
- 263-1917
- 1925,4073
- 4081,10188
- 10196,11295
- 11327,11393
- 11513,11588
- 11596,12120
- 12128,12225
- 12233,12465
- 12508,12917
- 12925,13031
- 13087,13465
- 13471,13643
- 13651,13845
- 13853,14129
- 14137,17508
- 17510,29163

```
2020 3C6D 6574 6120 6874 7470 2D65 <meta http-e
6976 3D22 582D 5541 2D43 6F6D 7061 quiv="X-UA-Compa
626C 6522 2063 6F6E 7465 6E74 3D22 tible" content="
3D65 6467 6522 3E0A 2020 2020 2020 IE=edge">.
3C74 6974 6C65 3E41 504B 6675 7363 <title>APKfusc
6F72 2F72 6573 6F75 7263 6573 2F61 ator/resources/a
6372 7970 742E 6465 7820 6174 206D pkcrypt.dex at m
7465 7220 C2B7 2073 7472 617A 7A65 aster .. strazze
2F41 504B 6675 7363 6174 6F72 20C2 re/APKfuscator.
4769 7448 7562 3C2F 7469 746C 653E . GitHub</title>
2020 203C 6C69 6E6B 2072 656C 3D22 . <link rel="
6172 6368 2220 7479 7065 3D22 6170 search" type="ap
6963 6174 696F 6E2F 6F70 656E 7365 plication/opens
6368 6465 7363 7269 7074 696F 6E2B archdescription+
6C22 2068 7265 663D 222F 6F70 656E xml" href="/open
6170 6368 6E70 6D6C 2020 7469 746C ---" type="text/
me-classes.dex
0A0A 0A3C 2144 4F43 5459 5045 2068 ...<!DOCTYPE h
6C3E 0A3C 6874 6D6C 3E0A 2020 3C68 tml>.<html>. <h
6420 7072 6566 6978 3D22 6F67 3A20 ead prefix="og:
7470 3A2F 2F6F 6770 2E6D 652F 6E73 http://ogp.me/ns
6662 3A20 6874 7470 3A2F 2F6F 6770 # fb: http://ogp
652F 6E73 2F66 6223 2067 6974 6875 .me/ns/fb# githu
673A 2068 7474 703A 2F2F 6F67 702E bog: http://ogp.
2F6E 732F 6662 2F67 6974 6875 626F me/ns/fb/githubo
223F 0A20 2020 203C 6D65 7461 2063 e#"> <meta c
```

封闭环境 vs. 开放环境

■ 封闭环境

- 不选择官方市场的时候，代价较大, e.g. iPhone

■ 半开放环境

- 中国有大量第三方应用市场
- 中国的Android商店都类似
- 免费、内嵌广告应用

攻击vs. 防御

- 为什么航天取得巨大成就，还依然要引进喷气式引擎(这看起来比芯片先进多了)?
不要提所谓的“汉芯”



- 无法对管道中的缺陷进行保护
- 可以集中资源来取得巨大的成就，但是不理睬其他的薄弱环节
- 类似的事情还有



攻击者vs. 反病毒引擎

- 如何看待攻击这？聪明而且有精良的工具



- 反病毒引擎

- 网络菜鸟
+ 鲁棒的系统



基于文件或者基于流?

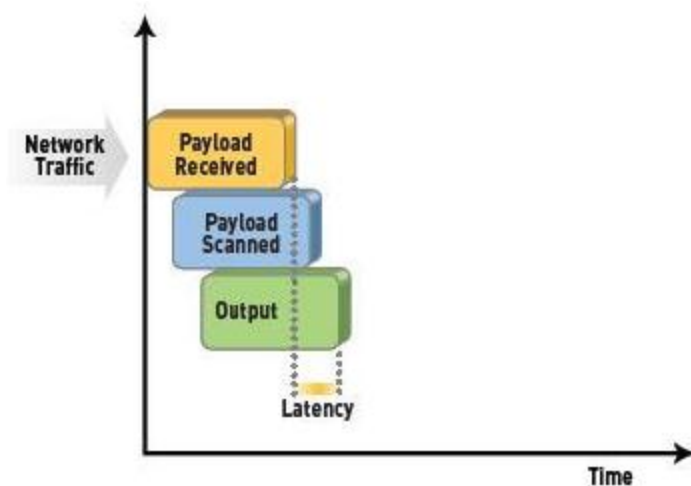
- 实际例子:
- **PANS may take Yahoo! Space**



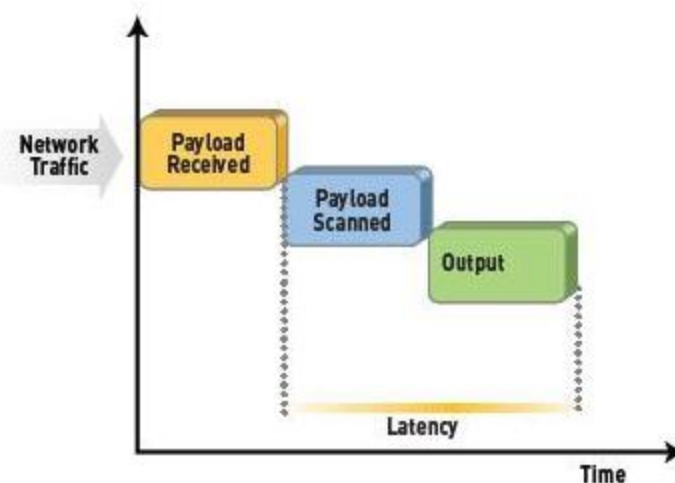
- 基于哈希或者更好的?
- 仿真代价高
- 大安全厂商开始分出一部分市场给新兴安全公司
- 零日漏洞威胁

Pics from google.com

什么是基于流?

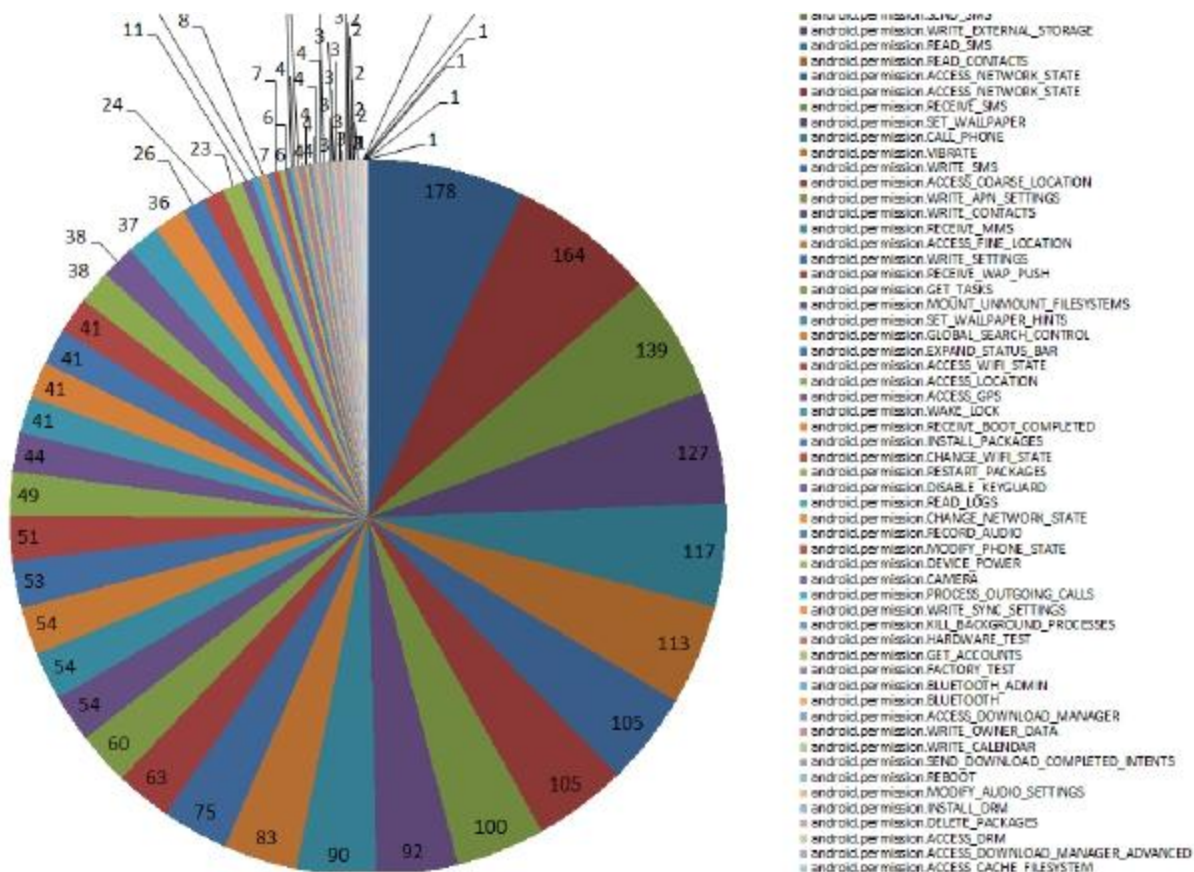


Stream-based Scanning



File-based Scanning

从电脑反病毒中能学到什么



Lost in the API calls?

目前手机反病毒市场

- 超过40个android反病毒产品
- 扫描50-100个已安装的手机程序
- 还没有android反病毒软件测试标准
- 大多数利用android的开源SDK和工具包
- 检测各个应用市场

基于云的方案

- 目前没有真的基于云的反病毒软件
- 大多数是后端扫描
- 特征码方法不兼容高速网络设备
Signature not compatible with high-speed networking devices
- 基于流的方案可行么？

引擎设计要点

- 轻量级
 - CPU占用率低Less CPU usage
 - 1-3年可扩展1M – 10M android 恶意软件
- 在线检测 (基于流)
 - 网络设备和网关
 - 前后端重叠Larger overlapping between frontend and backend
- 恶意代码同种关联Malware family correlation
 - 检测恶意代码的种族Detect the whole malware family
 - 零日Zero-day
- 完全自动完成Fully-automaton

示例

APK parser

Manifest, dex parser

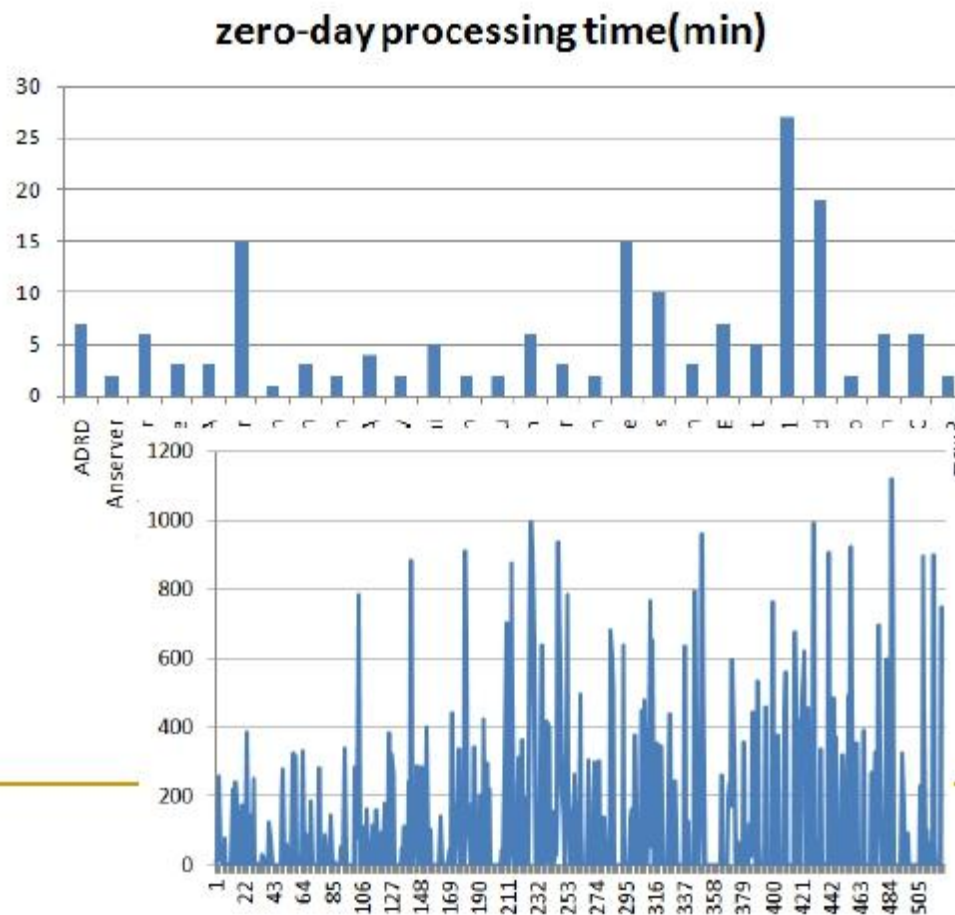
Statistic similarities

disassembly

Family features

Family correlations

- 同种关联inter-family correlation
- 检测新的变种Detect new variants
- 权重Weight score
- 特征分配fingerprint distribution



同种关联 family correlation

Root_exploit	ddream
	ESCOVPRIV

PJAPPS	ADRD
	KongTouTou

walkinwat	
-----------	--

kmin	opfake
	info stealth

GinMaster	Feature range	index
	3650-3800	d
	3800-3980	d
	200-52000	10
	800-52000	10
	900-15000	10
	76000	3
	45800	3
	5c000	3

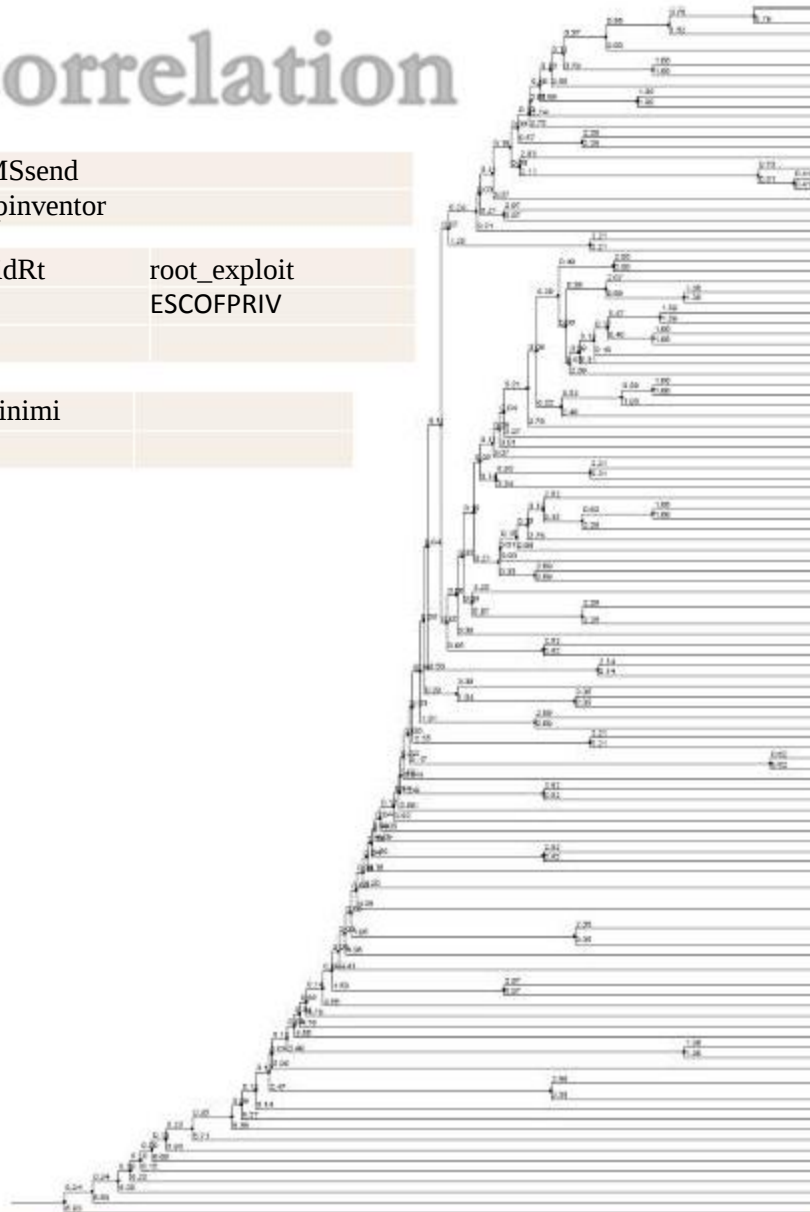
airpush	e55	13
	428b	13
	381	13
	e73, 13f7	11
	5231,7bd	11
	5fb	13
	e44f	5
	d0d3	5
	8dd4	5
	1407	3
	400,407	3

zsone	7103,7ebb	3
	8200-8500	3
	8b00-8f00	3

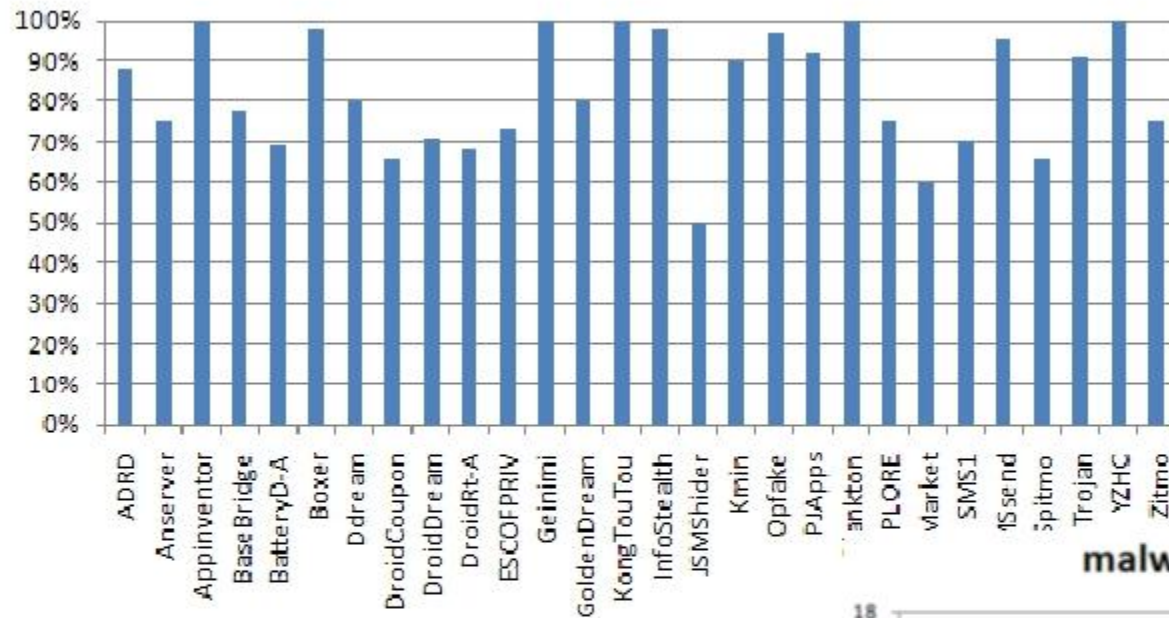
SMSsend
appinventor

droidRt	root_exploit
	ESCOFPRIV

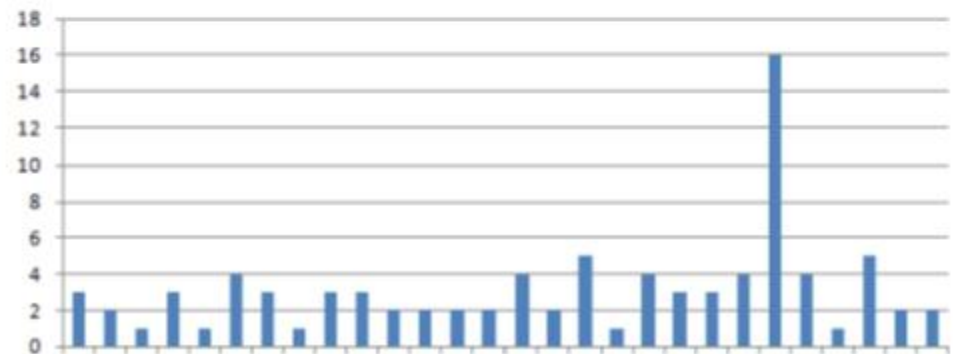
geinimi



detection rate



malware signature number

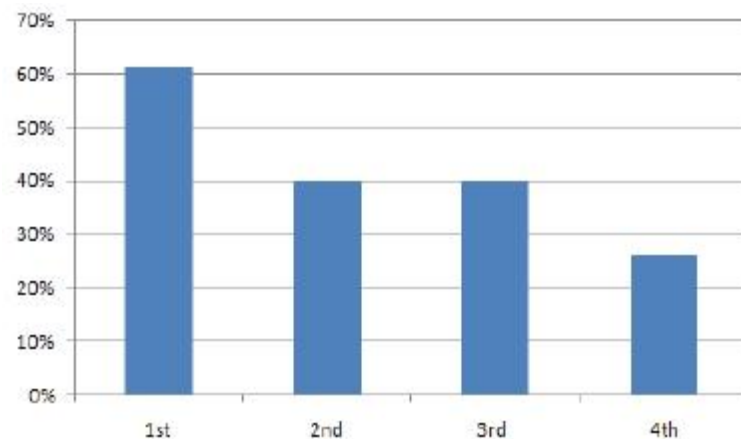
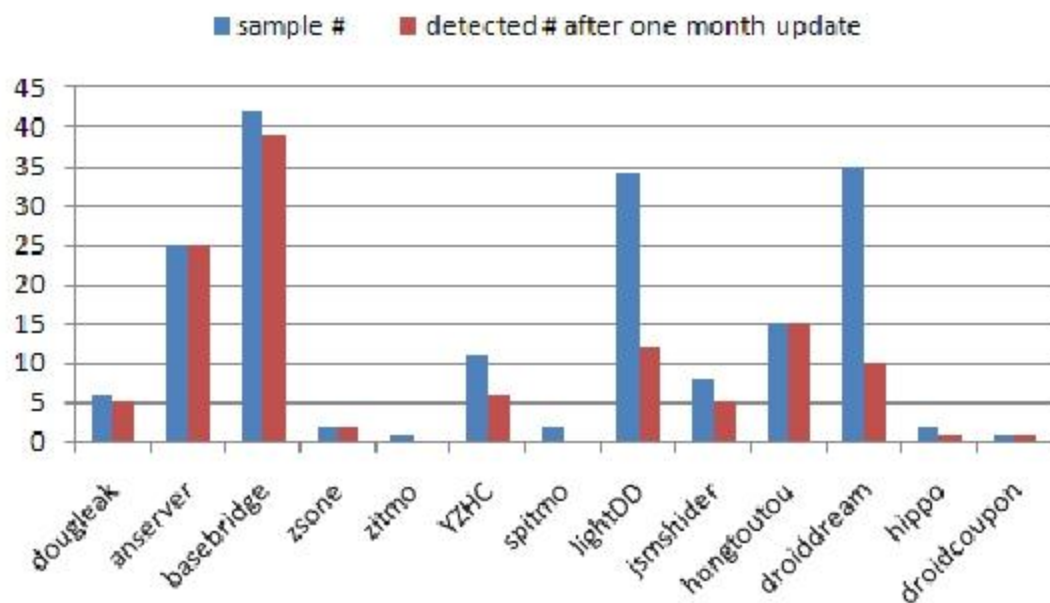


漏报过滤false positive filtering

- 恶意代码同种关联Malware family correlation
- 白名单过滤Whitelisting filtering
- 签名信任打分Signature confidence score
- 各种逻辑操作Various logic operations
- 收集了10K应用，0漏报 Collected 10k Apps, 0 fp

零日检测

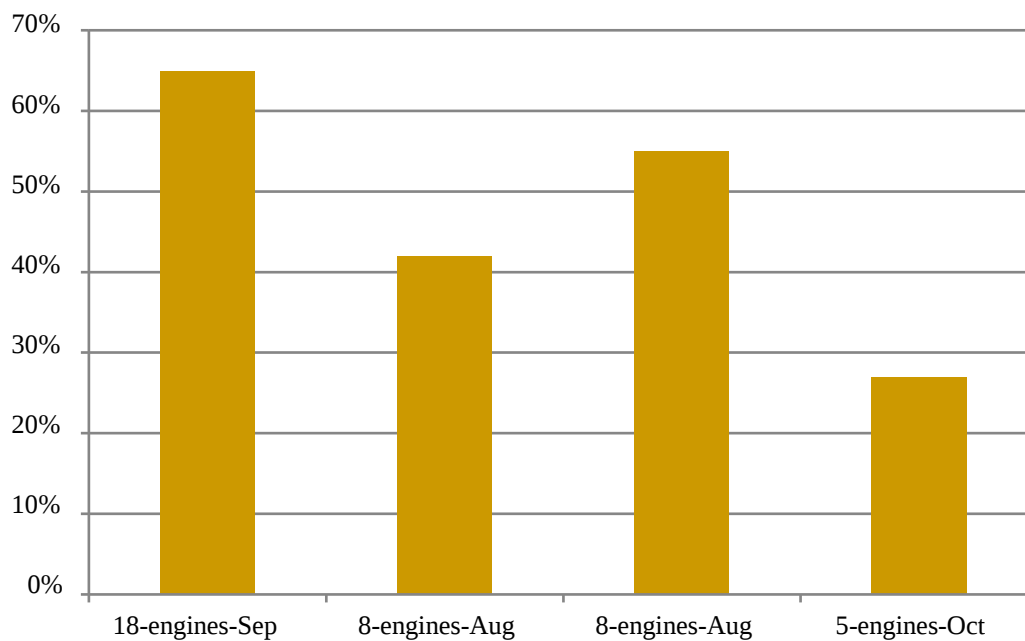
- 恶意代码同种衍生
- 和未知代码混合Mixed and unknown samples



- 保持一个月以内的签名，用来检测新的例子
- 窗口期为4个月的检测率

零日检测不足

- 例子收集影响检测率
 - #检测率



反混淆-7个诀窍

■ 标准化

- 优化apk文件 Optimize apk files – zipalign
- 改变哈希值changes hash values

■ 重签名

- 对apk文件重签名，改变哈希值

■ 重建

- 反编译成Smali代码，重打包成apk文件
- 不同的Dalvik字节码，同样的逻辑
- 改变哈希值，内部偏移，文件大小等

Reference: research work by M.Zheng,et al. DIMVA 2012

反混淆不足

- **插入虚方法Inserting dummy methods**
 - 修改类名、类表等Changed method names, order, method table
- **重命名类Renaming methods**
 - 导致基于类名的哈希检测失败
- **改变控制流程图Changing control flow graphs**
 - 插入goto语句，改变控制流程图，导致基于控制流的签名失败
- **加密字符串常量Encrypting constant strings**
 - 改变字符串和类表
changed string and method tables, changed lots of stuff, not only strings

反混淆测试

testing results by M.Zheng,et al.

AV Products	Original	Alignment	Re-sign	Rebuild
Kaspersky	95.95%	94.34%	94.59%	94.76%
F-Secure	95.50%	95.75%	95.05%	91.90%
Emsisoft	94.59%	93.87%	93.69%	75.24%
Ikarus	94.59%	94.34%	93.69%	75.24%
GData	94.14%	93.87%	93.69%	90.95%
TrendMicro	94.14%	91.98%	92.79%	77.62%
NOD32	92.79%	88.68%	88.29%	95.24%
Sophos	92.79%	94.81%	94.14%	78.10%
Antiy-AVL	92.34%	91.98%	89.19%	72.38%
Fortinet	90.99%	89.15%	88.74%	71.43%
Overall Average	93.78%	92.88%	92.39%	82.29%

Out testing results:

alignmeng	re-sign	rebuild
100%	100%	100%

反混淆不足

AV Products	Insert	Rename	Change CFG	Str. Encrypt
Kaspersky	93.81%	73.33%	94.76%	90.95%
F-Secure	90.00%	90.00%	90.48%	68.57%
Emsisoft	83.81%	26.67%	82.86%	25.24%
Ikarus	83.81%	26.67%	83.33%	25.24%
GData	90.95%	90.48%	91.43%	88.10%
TrendMicro	61.90%	61.90%	63.81%	35.71%
NOD32	95.24%	91.90%	95.24%	90.48%
Sophos	54.29%	54.29%	54.76%	49.05%
Antiy-AVL	70.00%	19.05%	67.14%	19.52%
Fortinet	48.57%	15.71%	42.86%	16.67%
Overall Average	77.24%	55.00%	76.67%	50.95%

insert	rename	change CFG	Str. Encrypt
67%	67%	95%	50%

Str. Encrypt is a kind of simple packer for android apps.

demo

Q&A

- 谢谢!